GFI Slides

Link to ISA101

ISA201 Lesson 4 Program Planning

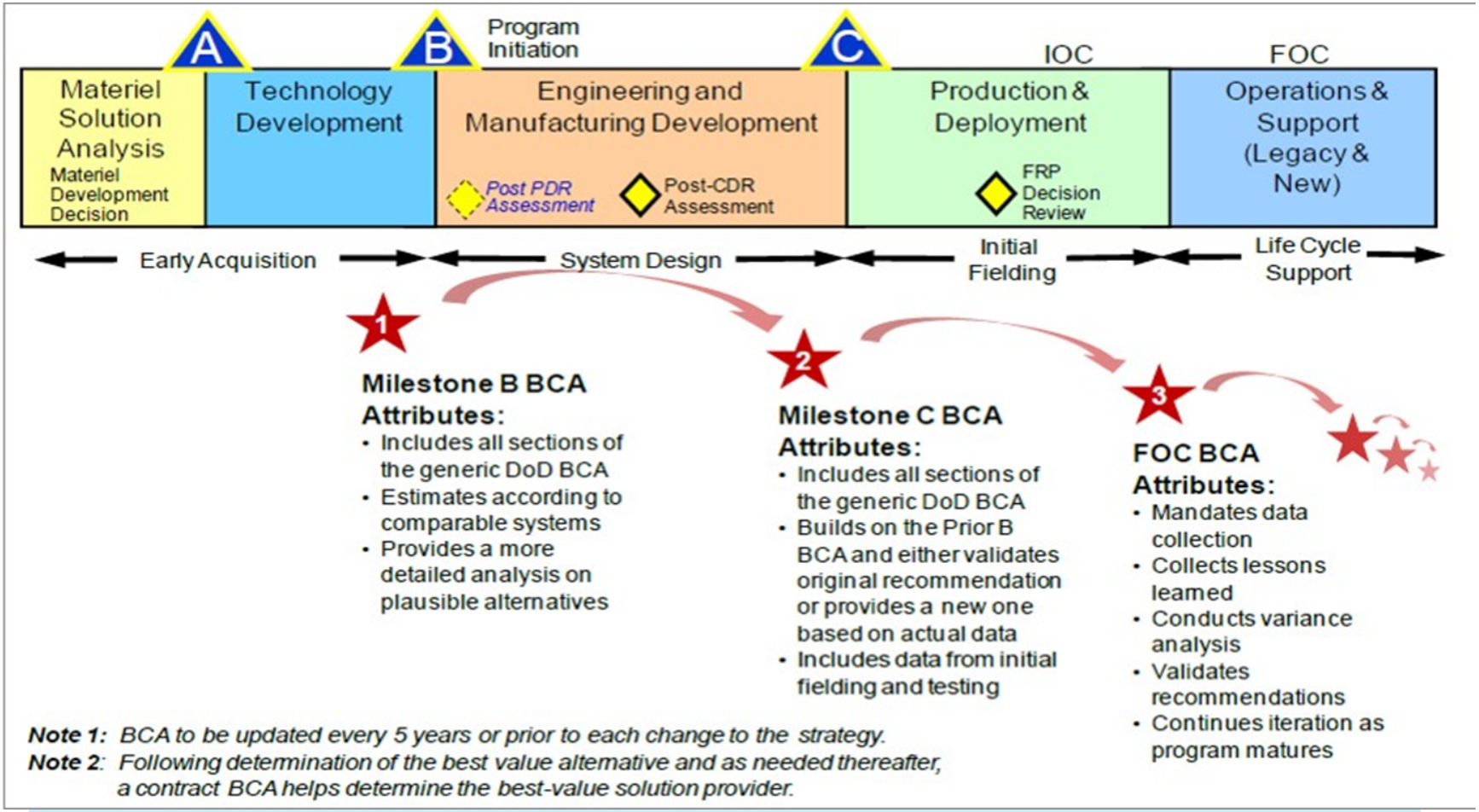
lesson 3.4

Comp Paper#4

Business Case Analysis (BCA)

- A formal, structured process
- A tool to assess business improvement alternatives
- Emphasis of the enterprise wide perspective of stakeholders and decision makers and assessment of the holistic effects impacted by the decision

When are BCA's Conducted?

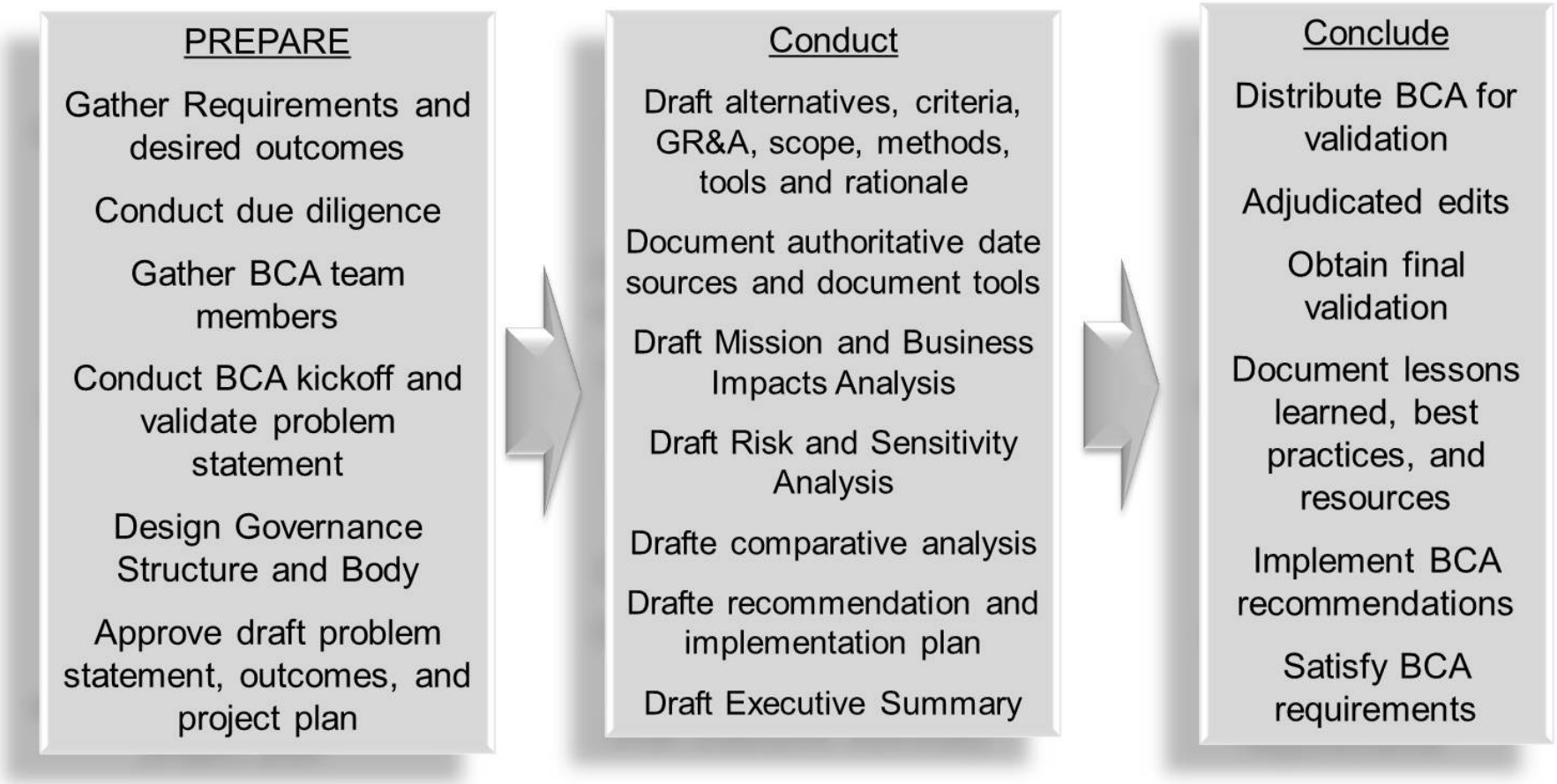

A
B
Program Initiation
C
IOC
FOC
Materiel Solution Analysis
Materiel Development Decision
Technology Development
Engineering and Manufacturing Development
Post PDR Assessment
Post-CDR Assessment
Production & Deployment
FRP Decision Review
Operations & Support (Legacy & New)
Early Acquisition
System Design
Initial Fielding
Life Cycle Support
1
Milestone B BCA Attributes:
• Includes all sections of the generic DoD BCA
• Estimates according to comparable systems
• Provides a more detailed analysis on plausible alternatives
2
Milestone C BCA Attributes:
• Includes all sections of the generic DoD BCA
• Builds on the Prior B BCA and either validates original recommendation or provides a new one based on actual data
• Includes data from initial fielding and testing
3
FOC BCA Attributes:
• Mandates data collection
• Collects lessons learned
• Conducts variance analysis
• Validates recommendations
• Continues iteration as program matures
Note 1: BCA to be updated every 5 years or prior to each change to the strategy.
Note 2: Following determination of the best value alternative and as needed thereafter, a contract BCA helps determine the best-value solution provider.

Product Support BCAs are accomplished throughout the life cycle.

BCA Major Elements

- BCA Purpose
  - Problem Statement, Objectives, Desired Outcomes, Requirements, Measurements
- BCA Process Components
  - Desired Outcomes and Requirements
  - Alternatives
  - Mission & Business Impacts
  - Risk Analysis & Mitigation Plans
  - Sensitivity Analysis
  - Recommendations
- BCA Quality Foundation
  - Governance, Research, Data Management, Due Diligence

Product Support BCA Process


PREPARE
Gather Requirements and desired outcomes
Conduct due diligence
Gather BCA team members
Conduct BCA kickoff and validate problem statement
Design Governance Structure and Body
Approve draft problem statement, outcomes, and project plan
Conduct
Draft alternatives, criteria, GR&A, scope, methods, tools and rationale
Document authoritative date sources and document tools
Draft Mission and Business Impacts Analysis
Draft Risk and Sensitivity Analysis
Drafte comparative analysis
Drafte recommendation and implementation plan
Draft Executive Summary
Conclude
Distribute BCA for validation
Adjudicated edits
Obtain final validation
Document lessons learned, best practices, and resources
Implement BCA recommendations
Satisfy BCA requirements

Tailoring of the process must occur to meet the needs of the stakeholders and sponsor.

SW Engineers Inputs to the Systems Engineer to support PS BCA Evaluations

These are the seven (7) design and support questions/criteria used when evaluating a potential Software Support Activity (SSA) in the PS BCA:

1) What are the projected costs to fix/update our proposed software design in O&S? Can we model the SSA costs using Parametric Modeling?
2) What is the SSA's proposed input into our SW developer's design plan with respect to Modular Open Systems Approach (MOSA)?
3) What data rights is the SSA proposing for our software design?
4) What tools does the potential SSA say they need to fix/update the software?
5) What is the potential SSA's data strategy with respect to data schema and overall data management?
6) What is the potential SSA's position on using secure coding practices?
7) What does the potential SSA recommend for our security architecture?

ISA201 Lesson 4 Program Planning

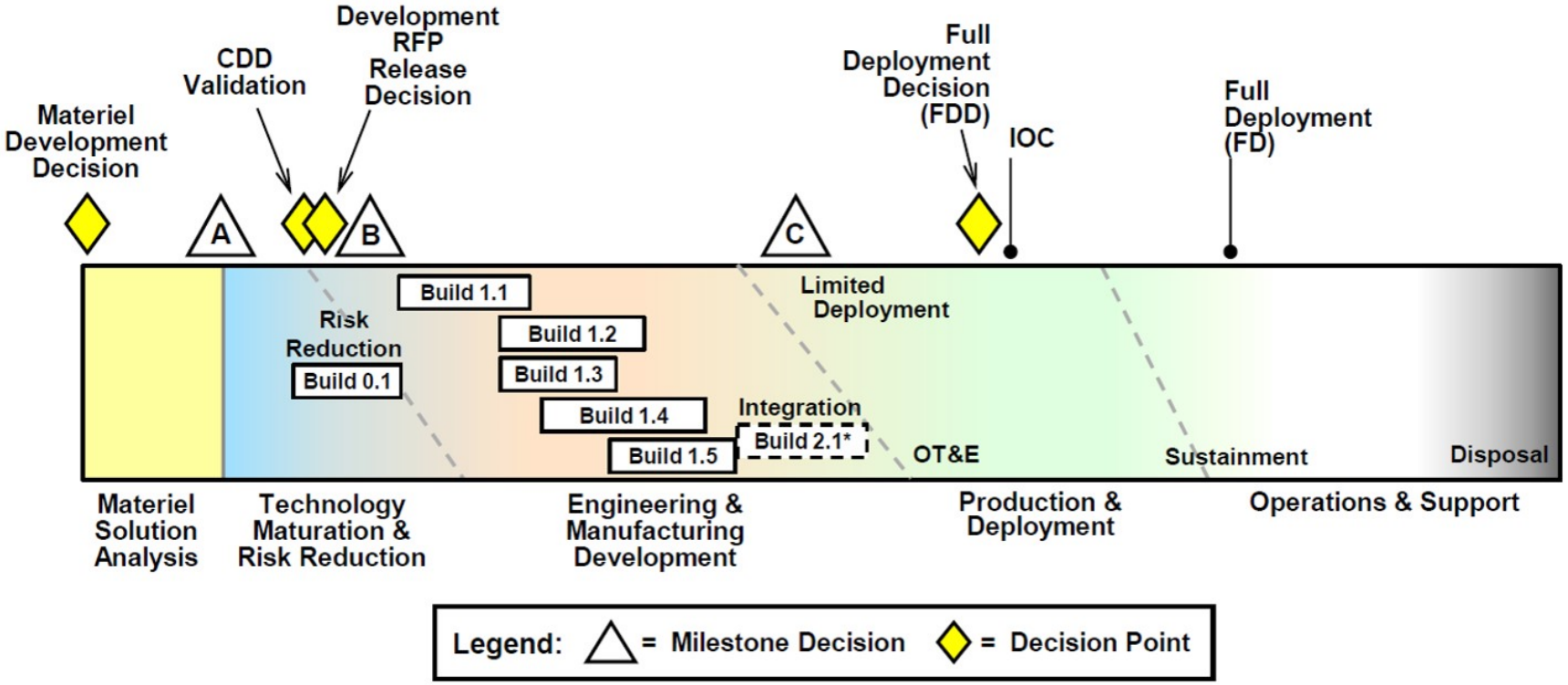
Link to ISA101 lesson 6.2 Comp Paper#2

Product-Tailored Acquisition Models (DODI 5000.02)

- Four basic models are tailored to the dominant characteristics of the acquisition
- The hybrids are described because many products will require combining models
  - Model 1: Hardware Intensive Program
  - Model 2: **Defense Unique Software Intensive Program**
  - Model 3: **Incrementally Deployed Software Intensive Program**
  - Model 4: Accelerated Acquisition Program
  - Model 5: **Hybrid Program A (Hardware Dominant)**
  - Model 6: **Hybrid Program B (Software Dominant)**

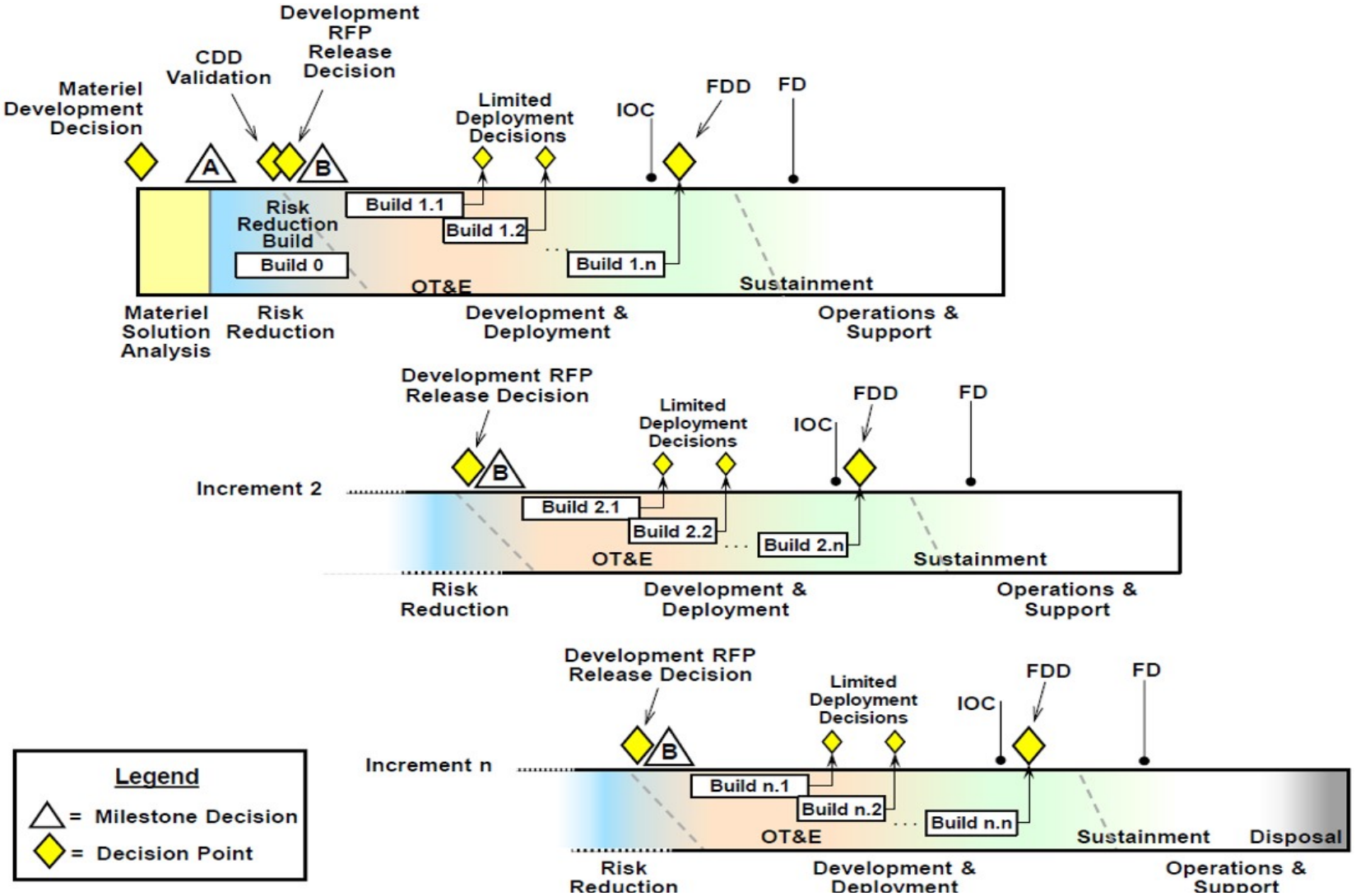
The models provide baseline approaches. A specific program should be tailored to the unique character of the product being acquired.

Defense Unique Software Intensive System (Model 2)

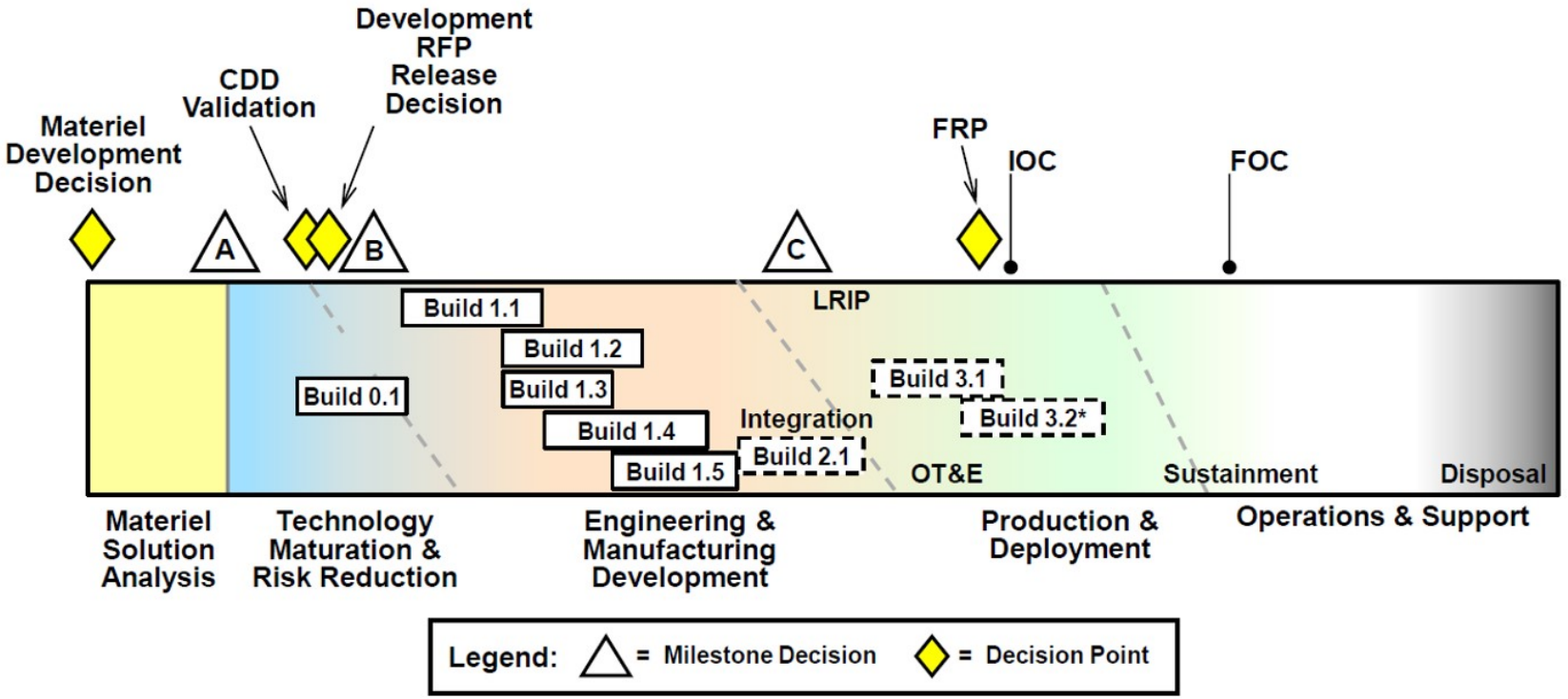


The number and type of builds will depend on system type.

Incrementally Deployed Software Intensive System (Model 3)

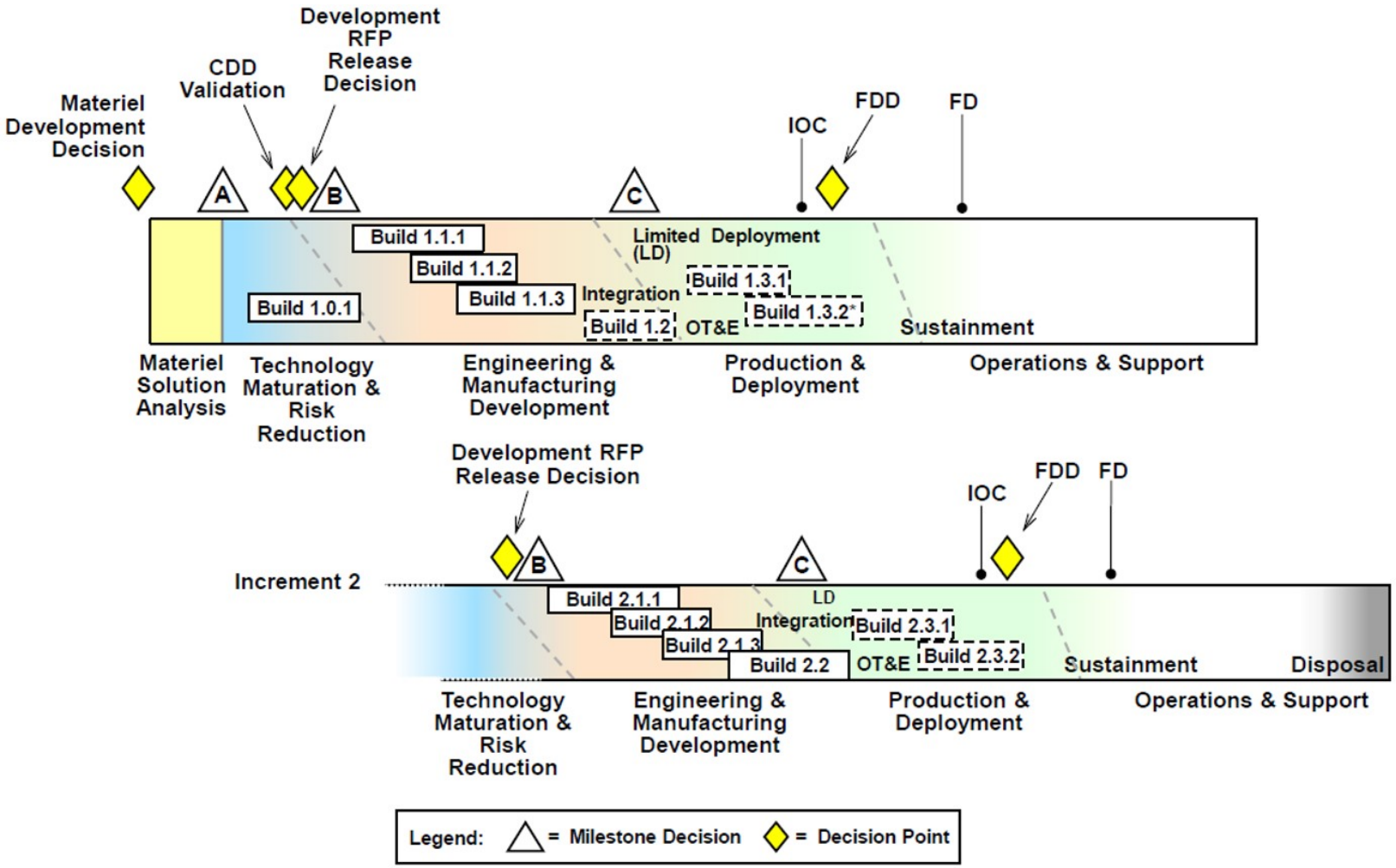


Hybrid Program A (HW Dominant) Model 5



* The actual number and type of builds during the program will depend on system type.

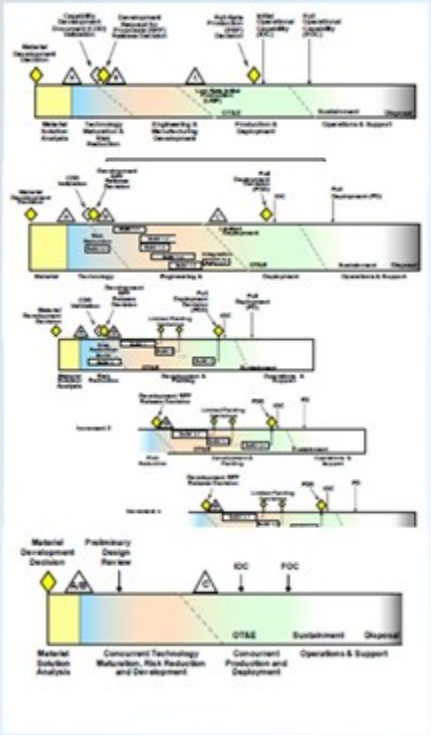
Hybrid Program B (SW Dominant) Model 6



Planning and DODI 5000.02

How to use the Document

What Model best accommodates the product I'm developing?



What business procedures apply to the program?

Materiel Development Decision

The Materiel Development Decision is based on a validated initial requirements document (an ICD or equivalent) and the completion of the AoA Study Guidance and AoA Study Plan. This decision directs execution of the AoA Study Guidance and AoA Study Plan, and authorizes the DoD Component to conduct the Materiel Solution Analysis Phase. This decision point is the entry point into the acquisition process for all defense acquisition programs; …

What statute and regulation is applicable to my program category (i.e., ACAT I –III) and milestone?

What detailed functional policy relates to my program?

Program Management, Systems Engineering, DT&E, OT&E, Sustainment, Human Systems, Affordability, AoAs, Resources and Cost, IT and Clinger-Cohen, Defense Business Systems, Urgent Operational Needs

ISA201 Lesson 13 Software Development

Link to ISA101 lesson 6.3 Comp Paper #10

Software Development Approaches Explained

Waterfall (Linear Sequential)

- Step by step development
- Can't begin one phase until previous phase is complete
- Can successfully use when requirements are well understood at the start with very little changes

Incremental

- A series of waterfall cycles
- Requirements known at the start and are divided into groups
- Identify a core set of functions for the first phase and build on it
- Best used with known requirements and when immediate functionality is needed

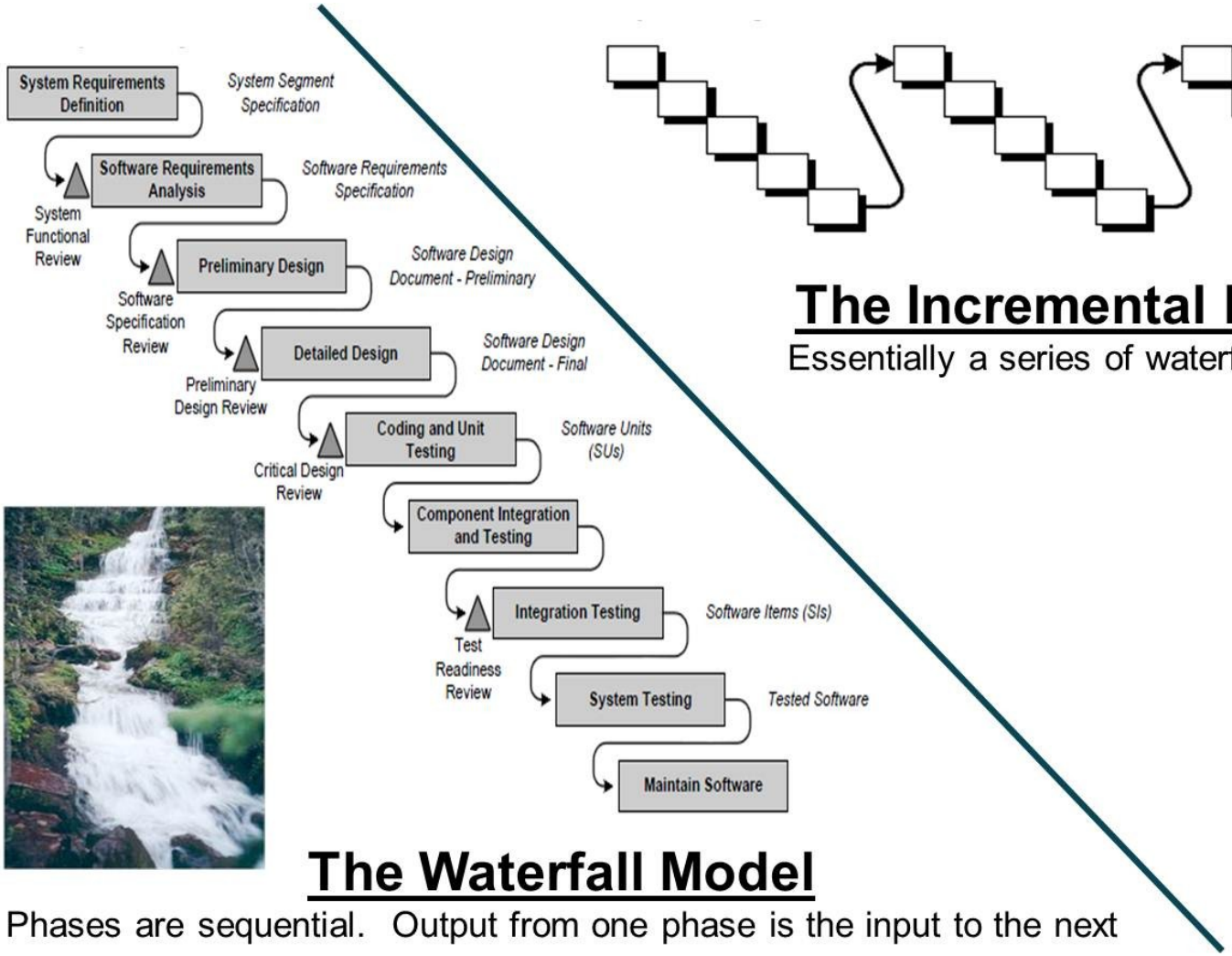
Spiral

- Can combine waterfall, evolutionary, and incremental
- Each spiral identifies high risk problems and develops solutions
- Each cycle ends in a review in which stakeholders agree on plans for the next cycle

Agile

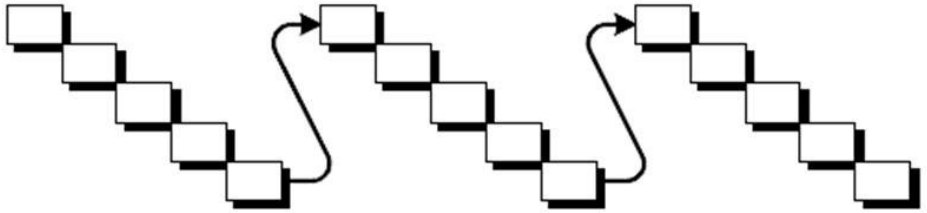
- A group of software development methods which adhere to the Agile Manifesto
- Emphasizes adaptability, customer collaboration and continuous delivery of valuable software

Software Development—Waterfall and Incremental



The Waterfall Model

Phases are sequential. Output from one phase is the input to the next



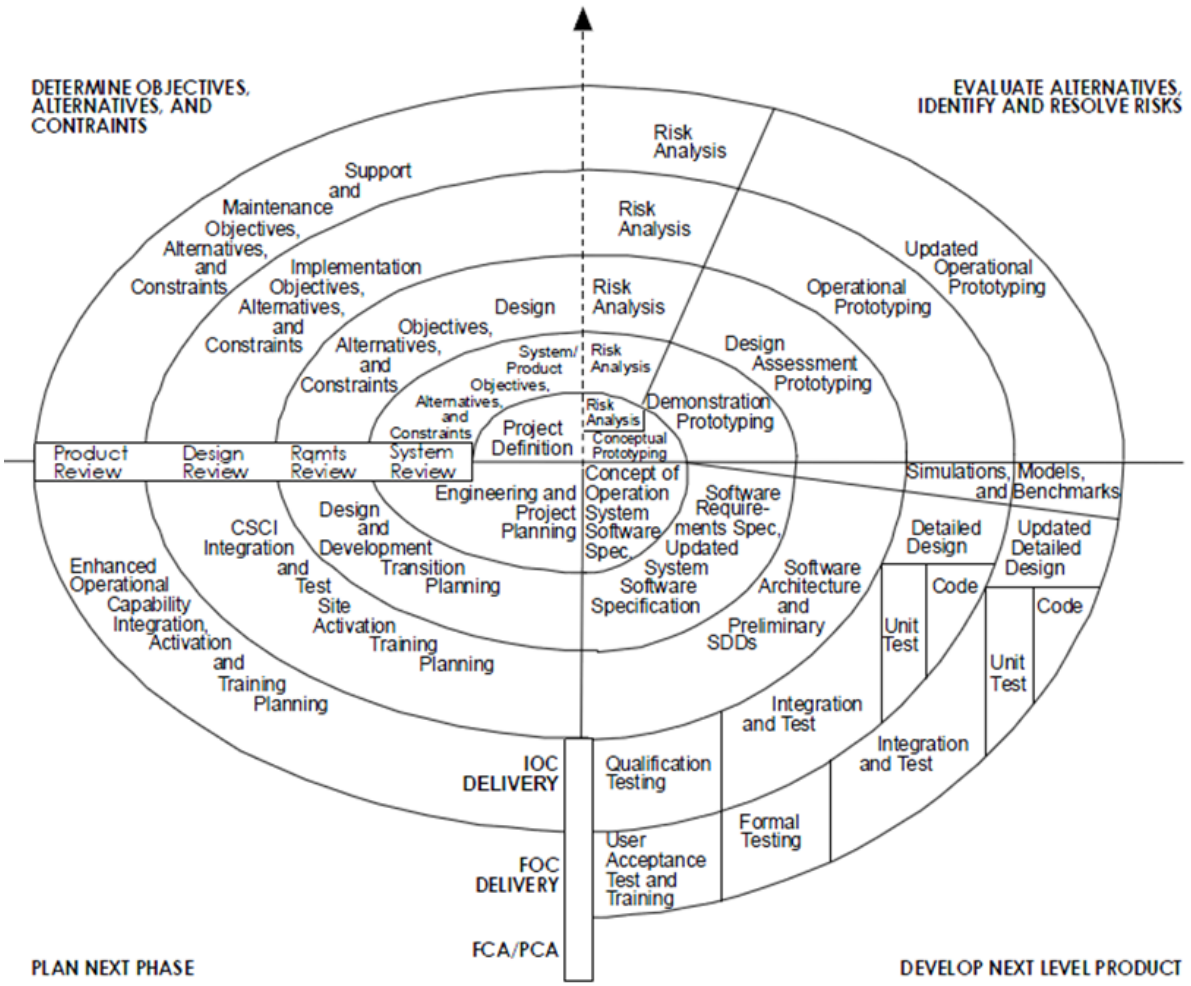
The Incremental Model

Essentially a series of waterfall cycles

Software Development—Spiral

Spiral Model

Designed to reduce technical risks and incorporate new technology and innovations. Requirements are “discovered” through development.



Software Development—Agile

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

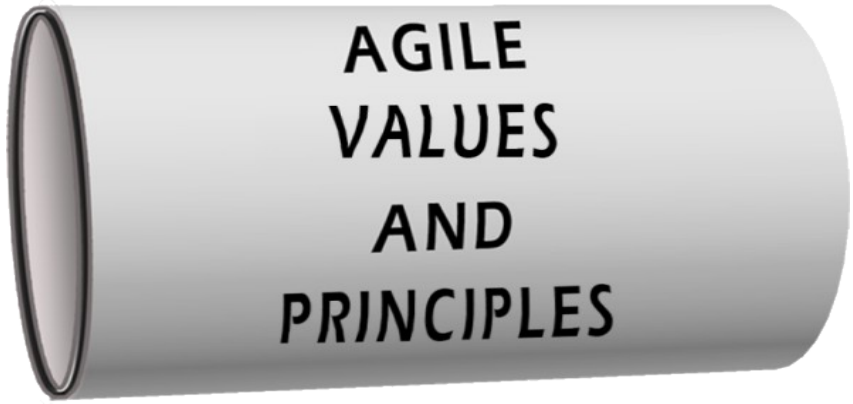
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

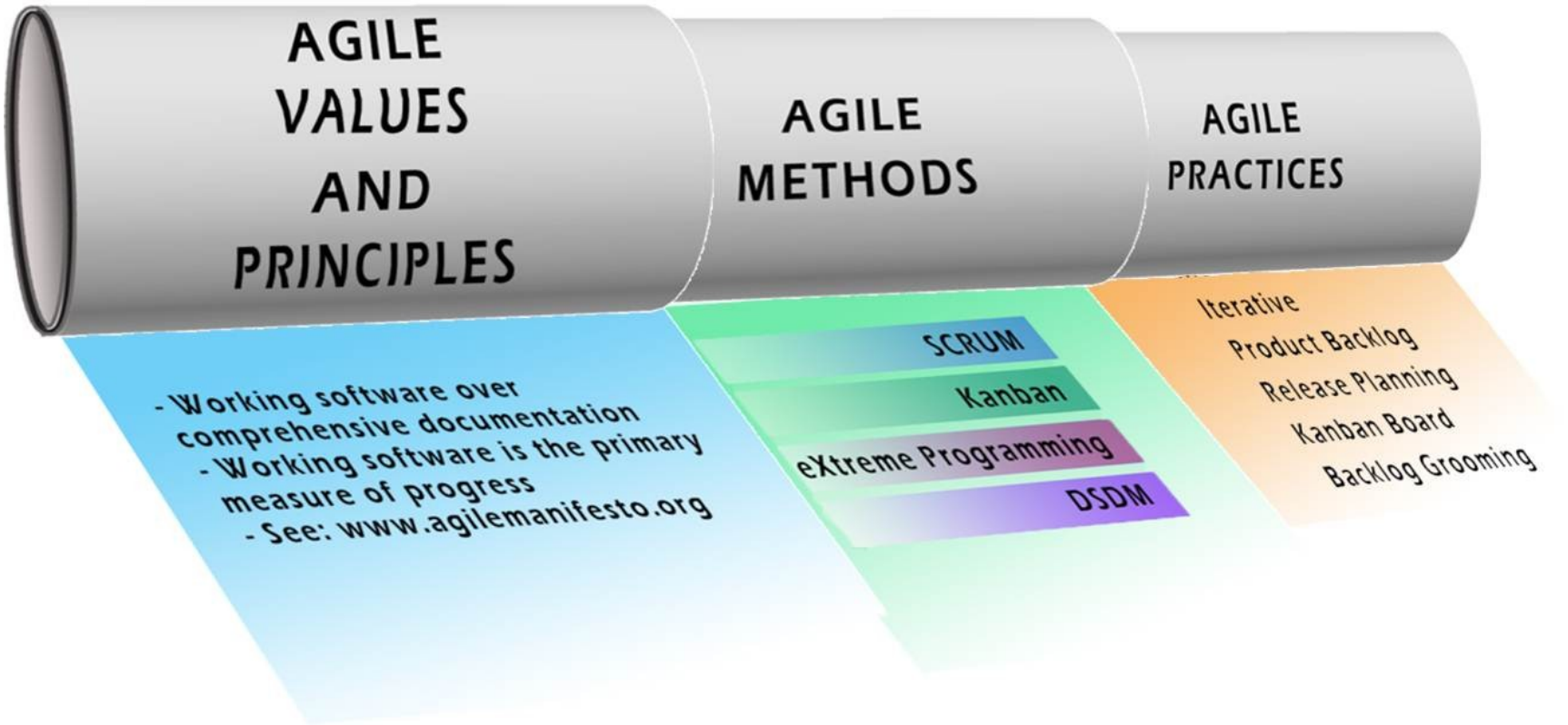
The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Individuals and interactions
over Processes & tools
Working software
over Comprehensive documentation
Customer collaboration
over Contract negotiation
Responding to change
over Following a plan

Agile Methods



Agile Practices

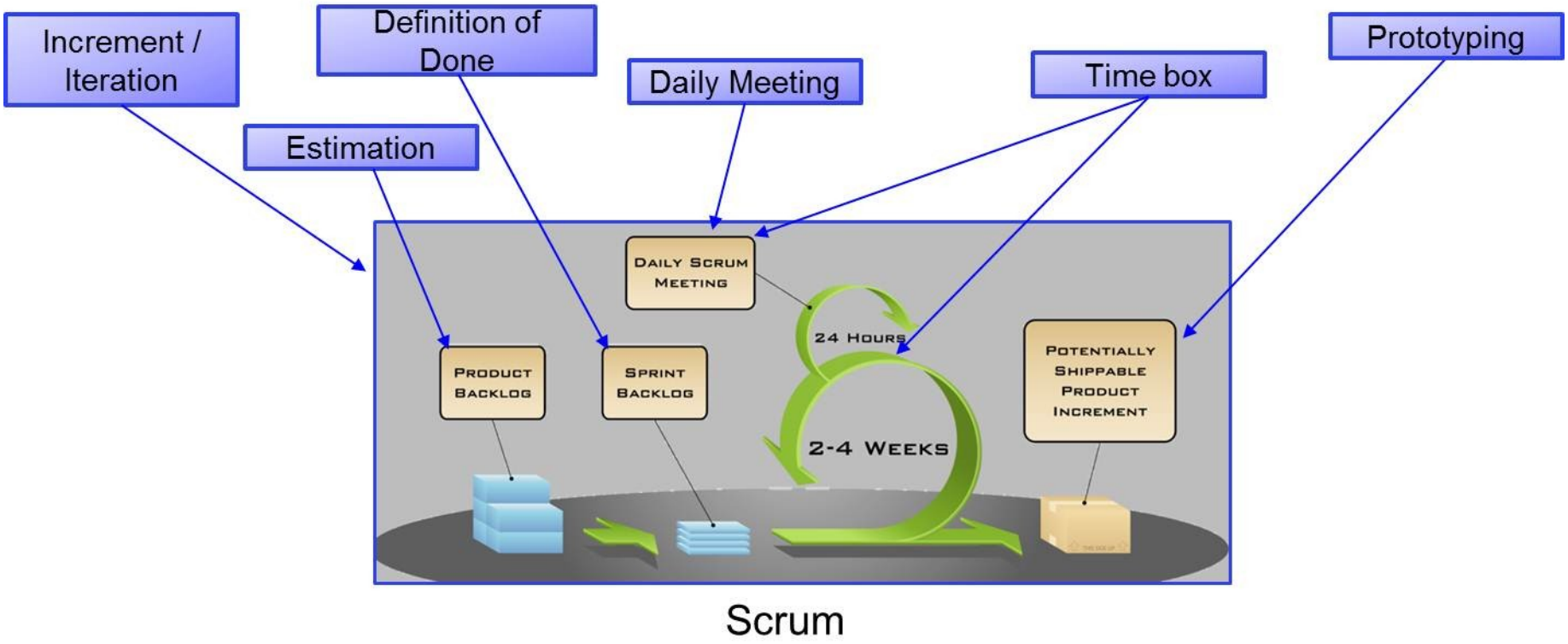
Time boxing	Unit testing
Increment	Continuous Integration
Iteration	User Stories
Definition of Done	Velocity
Product Backlog	Sustainable Pace
Release Planning	Prototyping
Estimation	Backlog Grooming
Daily Stand-up	*AND MANY MANY MORE*

Agile Practice—Time boxing

- Agile Practice: a previously agreed period of time during which a person or a team works steadily towards completion of some goal.
- Why: Parkinson's law—"work expands so as to fill the time available for its completion".
- Pre-Agile Examples: Pomodoro technique, Time-Constrained Project Management.

What is the purpose of Agile Methods?

- Put a subset of the practices together to use as a STARTING point!



Scrum

Software Development Approaches Advantages and Disadvantages

Incremental

- Advantages
  - Later development cycles can learn form previous cycles due to feedback
  - Risks spread over multiple cycles
  - Can test smaller portions of system easier
- Disadvantages
  - Must know most requirements at the start
  - Cost and schedule overruns may result in an unfinished system

Spiral

- Advantages
  - Improved Risk Management
  - Allows for better requirements definition
  - More responsive to user needs
- Disadvantages
  - More complex and harder to manage than other models
  - Development costs are higher
  - Schedule time is often increased

Waterfall

- Advantages
  - System is well documented
  - Phases correspond with project management phases
- Disadvantages
  - Deal with all risks in a single developmental effort
  - No working project until late
  - Minimal feedback

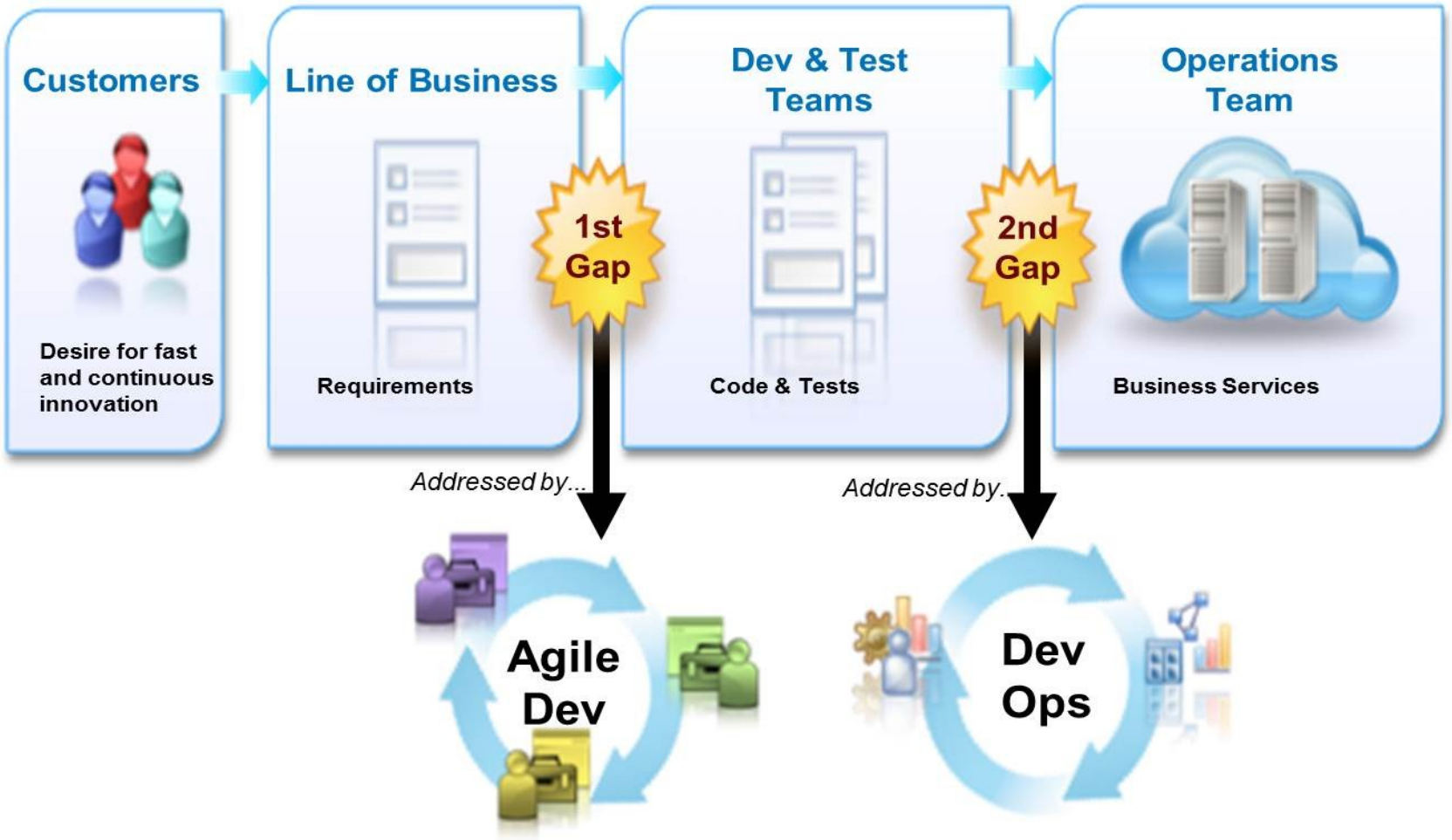
Agile

- Advantages
  - Early insight / buy-in by the stakeholders / users
  - Early course correction / “fail fast”
- Disadvantages
  - More dependence on tacit knowledge (e.g., less explicit documentation)
  - Dependence on availability of user/customers
  - Difficult aligning artifacts and measures with those of the larger traditional acquisition

Lesson Plan Status

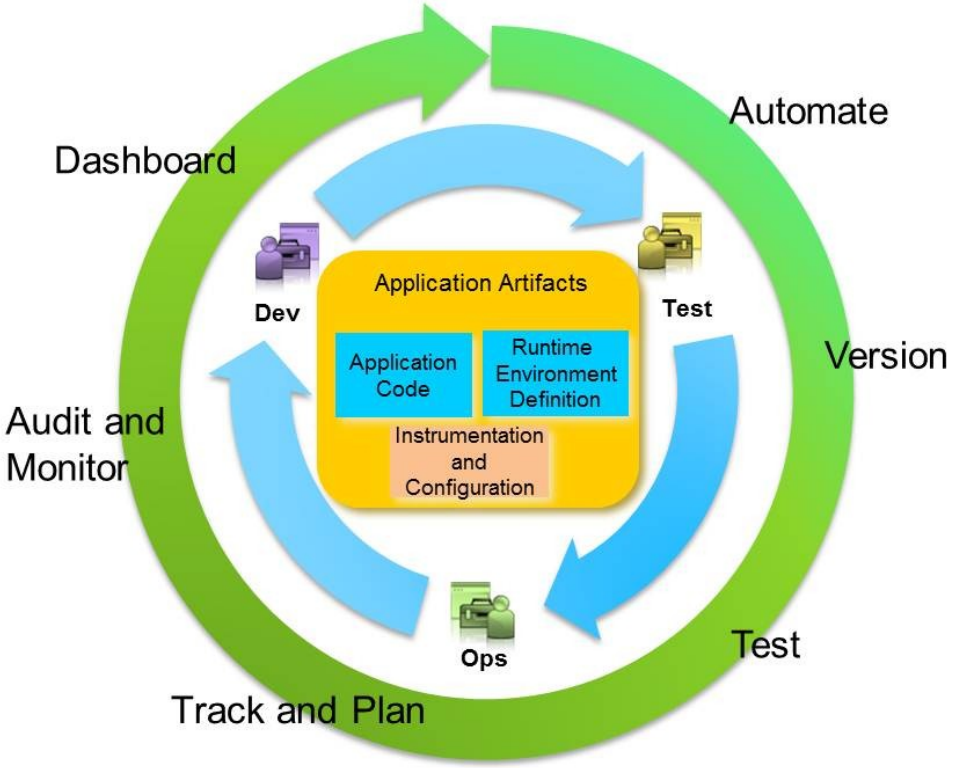
- Software Life Cycle Development Approaches
- DevOps
- Software Assurance
- Developer Process Maturity
- Integrating Commercial Off the Shelf (COTS) Products
- Exercise

Silos and Delivery Challenges



How to fix these problems?
Origin of “DevOps”

- Develop and test against production-like environments
- Iterative and frequent deployments using repeatable and reliable processes
- Continuously monitor and validate operational quality characteristics



Track and Plan everything
Version everything
Automate everything
Test everything
Audit and Monitor everything
Dashboard everything

Link to ISA101 lesson 10.2 and 11.4 Comp Paper#12

Process Assurance: Capability Maturity Model—Integration (CMMI)

- **Capability Maturity Model® Integration (CMMI)**
  - Was a collaborative effort sponsored by the Office of the Secretary of Defense/Acquisition and Technology (OSD/A&T) Systems Engineering with participation by government, industry, and the Software Engineering Institute (SEI)*.
  - Objective is to develop a product suite that provides industry and government with a set of products to support process and product improvement.
  - It can be used to guide process improvement across a project, a division, or an entire organization.
- **Benefits of implementing process improvement:**
  - The quality of a system is highly influenced by the quality of the process used to acquire, develop, and maintain it.
  - Process improvement increases product and service quality as organizations apply it to achieve their business objectives.
  - Process improvement objectives are aligned with business objectives.
  - CMMI maturity level can often be a good predictor of whether a software development project will incur cost and schedule overruns.

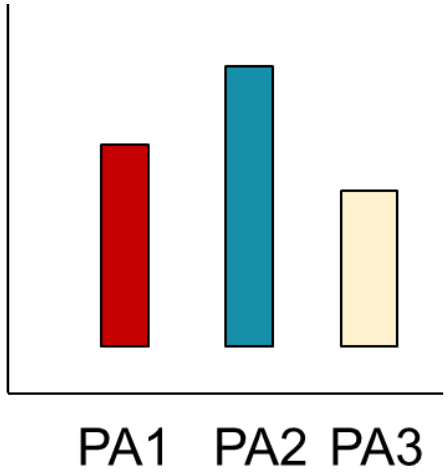
CMMI Institute
http://cmmiinstitute.com/

*CMMI is a spin-off of original SEI activity

Process Assurance: Capability Maturity Model—Integration (CMMI)

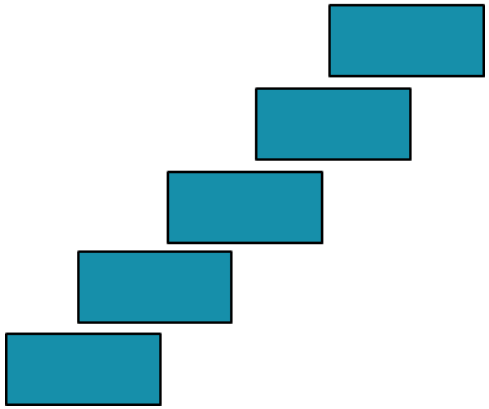
CONTINUOUS

Path enables organization to incrementally improve processes corresponding to an individual process area (PA) (or group of process areas) selected by the organization.



STAGED

Path enables organization to improve a set of related processes by incrementally addressing successive sets of process areas.



Process Assurance: Capability Maturity Model—Integration (CMMI)

Level	Continuous Representation Capability Levels	Staged Representation Maturity Levels
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4		Quantitatively Managed
Level 5		Optimizing

- Capability levels enable your organization to focus its process improvement efforts process area by process area from capability level 0 to capability level 3.
- Maturity levels provide a staging of processes for improvement across your organization from maturity level 1 to maturity level 5.

Process Assurance: More Quality Process Models and Standards

- ISO 9000, Quality Management and Quality Assurance Standards
  - Provides guidelines as to which document to use and how to use it.
  - Use of ISO 9001, 9002 or 9003 depends on business structure
- ISO 9001, Quality Systems
  - Model for Quality Assurance in Design/Development, Production Installation, and Servicing
- ISO 9002, Quality Systems
  - Model for Quality Assurance in Production and Installation
- ISO 9003, Quality Systems
  - Model for Quality Assurance in Final Inspection and Test
- ISO 9004, Quality Management and Quality Systems Element
  - Along with ISO 9000, ISO 9004 is an advisory document.

ISO 9001 gives the requirements that an organization must do to manage the **processes affecting quality** of its products and services

ISO 9000 - Quality management://www.iso.org/iso/iso_9000

ISA201 Lesson 10 Risk Management

Link to ISA101 lesson 12.3 Comp Paper#20

Identifying Risk Ideas: Technical Risk Drivers

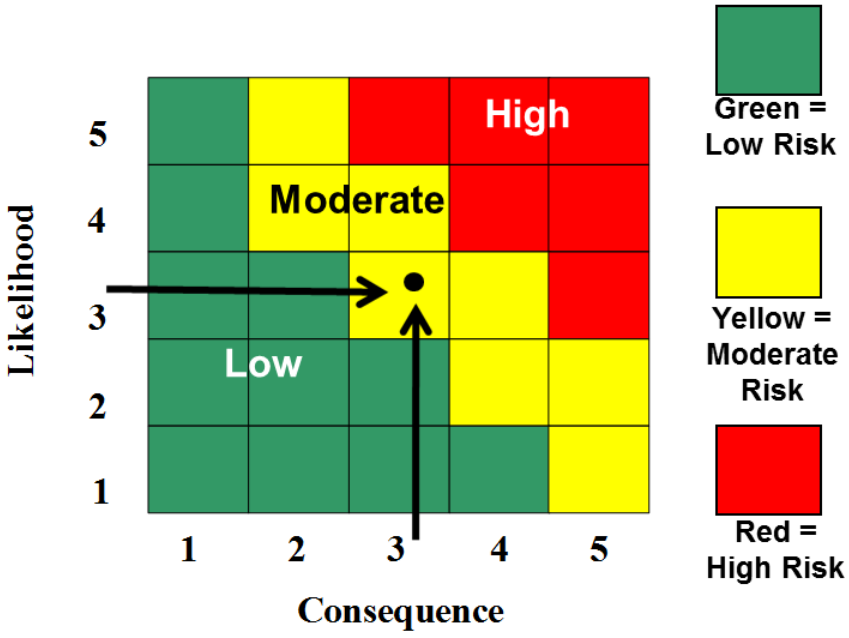
- Requirements
- Proprietary Data/Designs
- Technology
- Hardware State-of-the-Art
- Support Concepts
- Reliability and Maintenance
- Constraints
- Personnel
- Computer Resources
- Manufacturing Resources
- New Manufacturing/labs
- Standards
- Government Furnished Equipment/Personnel
- Environment
- Tools
- Data Rights
- Experience
- Documentation
- Management Approach

Software Specific Risks

- **Complexity**
- **Size**
- **Stability**
- **Developmental Approach**
- **Process Model**
- **Process Maturity**
- **Integration Approach**

How big is the risk and how likely is it to occur?

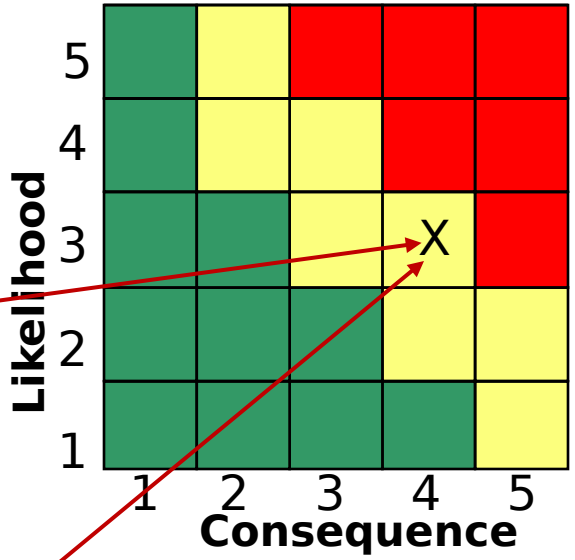
- Assess likelihood of occurrence
- Identify the possible consequences in terms of technical, schedule, and cost
- Assess consequence
- Identify the risk level in the 5X5 risk reporting matrix



Key Concept: Build the process **before** applying it!

Risk Analysis (Example)

Level	Likelihood	Probability of Occurrence
5	Near Certainty	> 80% to ≤ 99%
4	Highly Likely	> 60% to ≤ 80%
3	Likely	> 40% to ≤ 60%
2	Low Likelihood	> 20% to ≤ 40%
1	Not Likely	> 1% to ≤ 20%



Level	Technical Performance	Schedule	Cost
1	Minimal or no consequence to technical performance	Minimal or no impact	Minimal or no impact
2	Minor reduction in technical performance or supportability, can be tolerated with little or no impact on program	Able to meet key dates. Slip < 1 month	Budget increase or unit production cost increase (< 1% of Budget)
3	Moderate reduction in technical performance or supportability with limited impact on program objectives	Minor schedule slip. Able to meet key milestones with no schedule float. Slip 1 to 2 months	Budget increase or unit production cost increase (1 to 5% of Budget)
4	Significant degradation in technical performance or major shortfall in supportability; may jeopardize program	Program critical path affected. Slip >2 to 4 months	Budget increase or unit production cost increase (>5 to 10% of Budget)
5	Severe degradation in technical performance; Cannot meet KPP or key technical/supportability threshold; will jeopardize program success	Cannot meet key program milestones. Slip >4 months	Exceeds APB threshold (>10% of Budget)

Note: Underlined values must be tailored for specific programs or risks.

Risk Mitigation (formally Control)

- **Risk Mitigation** seeks to **actively** reduce risk to an acceptable level. Mitigation generally entails **taking action** to reduce the likelihood, and on occasion the consequence, of a risk to as low as possible in order to **minimize potential program impacts**.

- Risk mitigation plan implementation reduces level of program risks by:
  - **Directing** the teams to execute the defined and approved risk control plans
  - **Applying resources** (manpower, schedule, budget) to reduce the likelihood and/or consequence of risks
  - **Tracking** resource expenditure, technical progress, and risk impacts (benefits)
  - Outlining the risk reporting requirements for ongoing monitoring, to include “trip wires” which warrant elevating the risk to the next management level

Risk Mitigation Approaches

- Multiple Development Efforts: Create competing systems in parallel that meet the same performance requirements.
- Early Prototyping: Build and test prototypes early in the system development.
- Incremental Development: Defer capability to a follow-on increment.
- Technology Maturation: Use when a desired technology will replace an existing technology.
- Design of Experiments: Identify critical design factors that are sensitive, therefore potentially high risk, to achieve a particular user requirement.
- Open Systems, Standard Items, or Software Reuse: Select commercial specifications and standards or use. existing and proven hardware and software, where applicable.
- Models and Simulation: Investigate various design options and system requirement levels.

ISA201 Lesson 4 Program Planning

Link to ISA101 lesson 13.2 Comp Paper#26

Program Protection Planning (PPP)

Program Protection

- The integrating process for managing risks to DoD warfighting capability from foreign intelligence collection; from hardware, software, and cyber vulnerability or supply chain exploitation; and from battlefield loss throughout the system life cycle.

Program Protection Plan (PPP)

- Program managers will employ system security engineering practices and prepare a PPP to guide their efforts and the actions of others to manage the risks to critical program information and mission-critical functions and components associated with the program.
- The PPP will be submitted for MDA approval at each Milestone review, beginning with Milestone A.
- Program managers will describe in their PPP the program's Critical Program Information and mission-critical functions and components; the threats to and vulnerabilities of these items; the plan to apply countermeasures to mitigate associated risks; and planning for exportability and potential foreign involvement.

The Cybersecurity Strategy is required per statute as an appendix to the PPP

Program Protection Plan Contents

- Plan to align PPP and Prime Contractor Program Protection Implementation Plan(s)
- Critical Program Information (CPI) and Critical Functions and Components Protection
  - Identification Methodology, Inherited CPI and Critical Components, and Organic CPI and Critical Components
- Assurance that critical defense technologies, to include CPI, associated with more than one Research, Development, and Acquisition (RDA) program, are protected to the same degree by all involved DoD activities
- Identified threats and vulnerabilities to CPI and critical functions/components

PPP Outline & Guidance v 1.0 July 2011

Program Protection Plan Contents

- Other System Security-Related Plans & Documents (SEP, TEMP)
- How Program Protection risks will be integrated with overall Program risk management
- Foreign Involvement, Defense Exportability Features
- Processes for Management and Implementation of PPP
- Processes for Monitoring and Reporting Compromises
- Summarize the plan/procedure for responding to a CPI compromise or a supply chain exploit
- Program Protection Costs

Program Protection Plan Appendices

- Security Classification Guide (SCG)
  - The SCG may be referenced or pointed to rather than included in the document.
- Counterintelligence Support Plan (CISP)
  - The CISP may be referenced or pointed to rather than included in the document.
- Criticality Analysis
  - The results of the most recent Criticality analysis must be included and should be updated regularly
- Anti-Tamper Plan
  - Not all programs will require an Anti-Tamper plan. If an Anti-Tamper Plan is required, use the template developed by the Executive Agent for Anti-Tamper.
- Acquisition Cybersecurity Strategy
  - See next slide for content

*Cybersecurity Strategy Outline

I. Program and System Description
II. Information Assurance/ Cybersecurity Requirements
III. System IA/Cybersecurity Approach (high level)
IV. Acquisition of IA/Cybersecurity Capabilities and Support
V. System Certification and Accreditation (Assessment and Authorization (A&A))
VI. IA/Cybersecurity Testing
VII. IA/Cybersecurity Shortfalls
VIII. Policy and Guidance
IX. Point of Contact

The Acquisition Documentation Streamlining effort requires this to be an appendix to the PPP
Cybersecurity threats must be included in the PPP threat table

* DIACAP and previous policy refers to this as the IA Strategy. Official DoD guidance for the Cybersecurity Strategy is expected soon. Use this and change DIACAP language to RMF until guidance is finalized.

ISA201 Lesson 12 Systems Design

Link to ISA101 lesson 13.2 (safety) Comp Paper#22

Software Safety

- Both a subset of Software Development and Software System Safety
- The process of identifying Safety-Critical Software and assessing the criticality
- A series of requirements, design, and verification activities used to demonstrate that software will meet the customer's safety requirements to an acceptable level of confidence

This slide and the following 5 slides on Software Safety were adapted from "Software Safety Tutorial". The 27th International System Safety Conference, Presented by Josh McNeil and Willie Fitzpatrick, PhD, August 2009

Software Safety Failure Example

The British destroyer H.M.S. Sheffield was sunk in the Falkland Islands war. According to one report, the ship's radar warning systems were programmed to identify the Exocet missile as "friendly" because the British arsenal includes the Exocet's homing device and allowed the missile to reach its target, namely the Sheffield.

Software Safety Failure Example

- The Ariane 5 satellite launcher malfunction was caused by a faulty software exception routine resulting from a bad 64-bit floating point to 16-bit integer conversion

Software Safety Failure Example

An Iraqi Scud missile hit Dhahran barracks, leaving 28 dead and 98 wounded. The incoming missile was not detected by the Patriot defenses, whose clock had drifted 0.36 seconds during the 4-day continuous siege, the error increasing with elapsed time since the system was turned on. This software flaw prevented real-time tracking. The specifications called for aircraft speeds, not Mach 6 missiles, for 14-hour continuous performance, not 100. Patched software arrived via air one day later.

Software Safety Failure Example

- Aegis was installed on the U.S.S. Ticonderoga, a Navy cruiser. After the Ticonderoga was commissioned the weapon system underwent its first operational test. In this test it failed to shoot down six out of 16 targets because of faulty software; earlier small-scale and simulation tests had not uncovered certain system errors. In addition, because of test-range limitations, at no time were more than three targets presented to the system simultaneously. For a sizable attack approaching Aegis' design limits the results would most likely have been worse.

Software Safety Definitions

- **Software System Safety**—The optimization of system safety in the design, development, use, and maintenance of software systems and their integration with safety-critical hardware systems in an operational environment.
- **Safety-Critical**—A term applied to a function, condition, event, operation, process, component, or other element of a system whose proper recognition, control, performance or tolerance is essential to safe system operation or use (e.g., safety-critical function, safety-critical path, safety critical component).
- **Safety-Critical Software Function (SCSF)**—A function, whose proper execution is necessary to reduce the risk of a hazard OR if performed incorrectly, performed out-of sequence, or not performed could contribute to a hazard, compromise a safety mitigation feature, or result in a diminished level of safety.
- **Mishap**—An unplanned event or series of events resulting in death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment.
- **Hazard**—Any real or potential condition that can cause:
  1. Injury, illness, or death to personnel;
  2. Damage to or loss of a system, equipment or property; or

Software Safety Ground Rules

- **Software is a hazard cause**
  - Software is a causal factor or contributes to a hazard
  - Two types of hazardous functions
    - Mitigating or “Must Work” functions (e.g. flight controls, …)
    - Contributing or “Must Not Work” functions (e.g., premature missile firing, …)
- **Software hazard contributors** result from inadequate System and Software design and requirements
  - Caused by lack of design foresight, system development errors, inadequate requirements specifications, …
  - Hardware failures can modify intended software behavior
- Traditional **Independent Verification & Validation (IV&V**) does not address Software System Safety
  - IV&V is focused on all mission critical software functions and not traditionally focused on safety-critical software functions
  - Software System Safety focuses on safety-critical software functions as hazard causes

Safety of Software

Software System Safety

System Safety and Risk Assess Activities

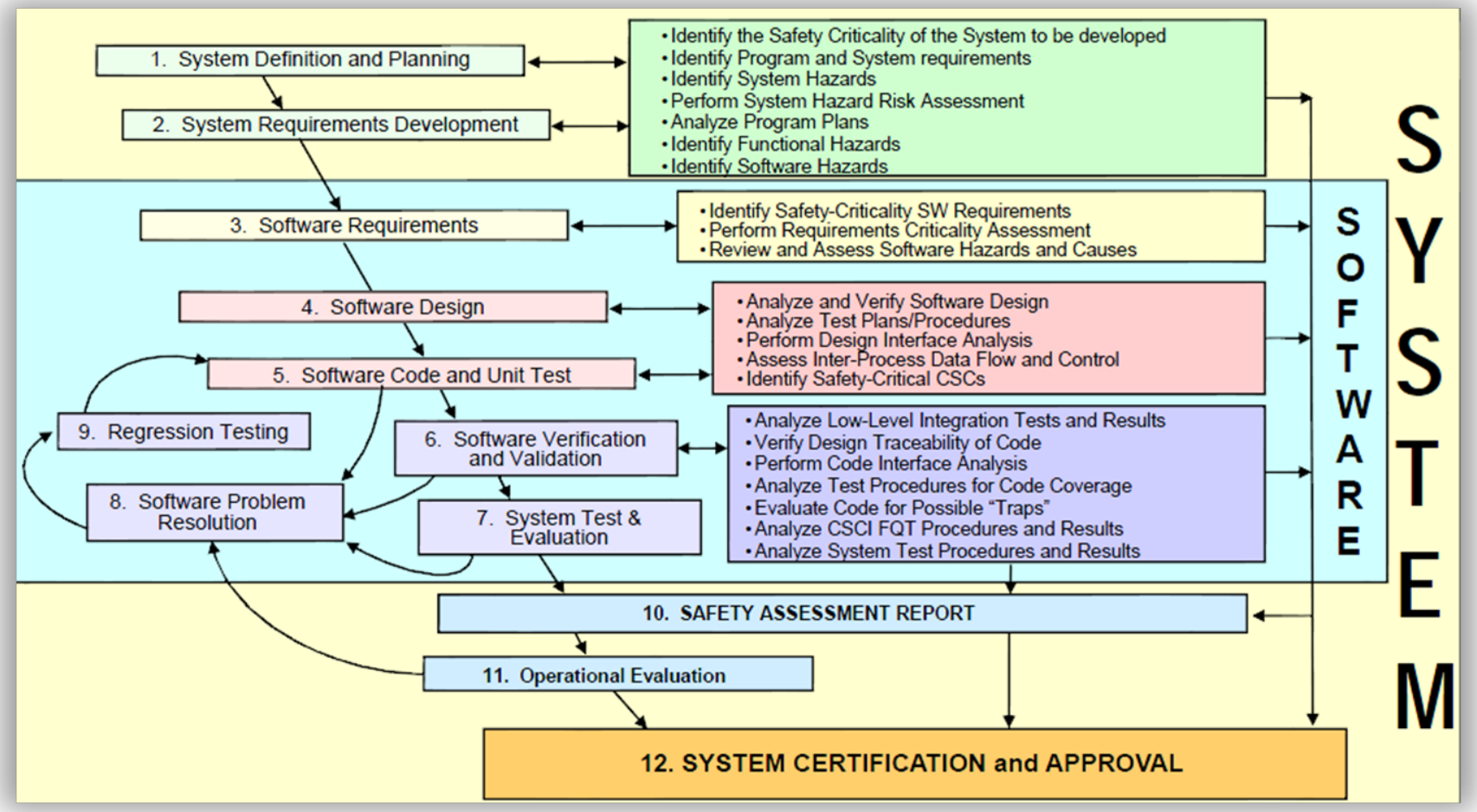
- System Safety Analyses to include software & hazard reports
- Perform hazard risk assessments
- Identify System Safety requirements Identification of safety critical software functions, safety-critical software requirements (SCRs), and associated criticality
- Identification of risk mitigation and perform risk assessment
- Produce Safety Assessment Report

Software Safety

Software Safety Development Assurance Activities

- Participate in Hazard Analysis to identify specific S/W causes and faults
- Assist in identification of safety-critical software functions, SCRs, & criticality
- Flag SCRs & criticality in requirements traceability database
- Trace requirements to design & code
- Perform development assurance analyses and tests to ensure software meets (Level of Rigor) LOR & (Safety Integrity Level) SIL
- Provide V&V evidence to System Safety for final risk assessment

Software System Safety Analysis


SYSTEM AND SOFTWARE DEVELOPMENT LIFECYCLE
SYSTEM AND SOFTWARE SAFETY CONTRIBUTIONS
1. System Definition and Planning
2. System Requirements Development
3. Software Requirements
4. Software Design
5. Software Code and Unit Test
6. Software Verification and Validation
7. System Test & Evaluation
8. Software Problem Resolution
9. Regression Testing
10. SAFETY ASSESSMENT REPORT
11. Operational Evaluation
12. SYSTEM CERTIFICATION and APPROVAL
• Identify the Safety Criticality of the System to be developed
• Identify Program and System requirements
• Identify System Hazards
• Perform System Hazard Risk Assessment
• Analyze Program Plans
• Identify Functional Hazards
• Identify Software Hazards
• Identify Safety-Criticality SW Requirements
• Perform Requirements Criticality Assessment
• Review and Assess Software Hazards and Causes
• Analyze and Verify Software Design
• Analyze Test Plans/Procedures
• Perform Design Interface Analysis
• Assess Inter-Process Data Flow and Control
• Identify Safety-Critical CSCs
• Analyze Low-Level Integration Tests and Results
• Verify Design Traceability of Code
• Perform Code Interface Analysis
• Analyze Test Procedures for Code Coverage
• Evaluate Code for Possible "Traps"
• Analyze CSCI FQT Procedures and Results
• Analyze System Test Procedures and Results
SOFTWARE
SYSTEM

Software Safety Lessons Learned

- Most Software Safety problems come from vague or incomplete requirements that are misunderstood or misinterpreted.
- Legacy/Reused software must be included in Software System Safety Analyses
- Perform Software System Safety Analyses early in program lifecycle
- Safety-Critical Software requirements derived from Software System Safety Analyses must be included and identified as Safety-Critical in the Software Requirement Specification (SRS)

Software Safety Lessons Learned (Continued)

- Correct implementation of software exception handlers MUST BE VALIDATED (exceptions are runtime errors with undesired results or that disrupt normal program flow—handlers are built to assist the system to safely counteract the exception/error)
- Safety concerns from Software System Safety Analysis must be addressed in Software Test Plans and Procedures (Boundary value testing, Full integration end-to-end testing)
- Safety-Critical software must undergo full integration end-to-end testing, all software must be functioning and not inhibited

ISA201 Lesson 08 Cybersecurity

Link to ISA101 lesson 13.7 Comp Paper#26

Risk Management Framework (RMF) for DoD Information Technology (IT)

- DoD Instruction 8510.01
  - Risk Management Framework (RMF) for DoD Information Technology (IT)
  - Signed March 12, 2014
- More consistent with established disciplines and best practices for effective systems engineering, systems security engineering, and program protection planning outlined in DoDI 5000.02
- Leverages and builds upon numerous existing federal policies and standards so we have less DoD policy to write and maintain.

DoD participates in CNSS and NIST policy development as a vested stakeholder with the goals of a more synchronized cybersecurity landscape and to protect the unique requirements of DoD Missions and warfighters

Key RMF Documents

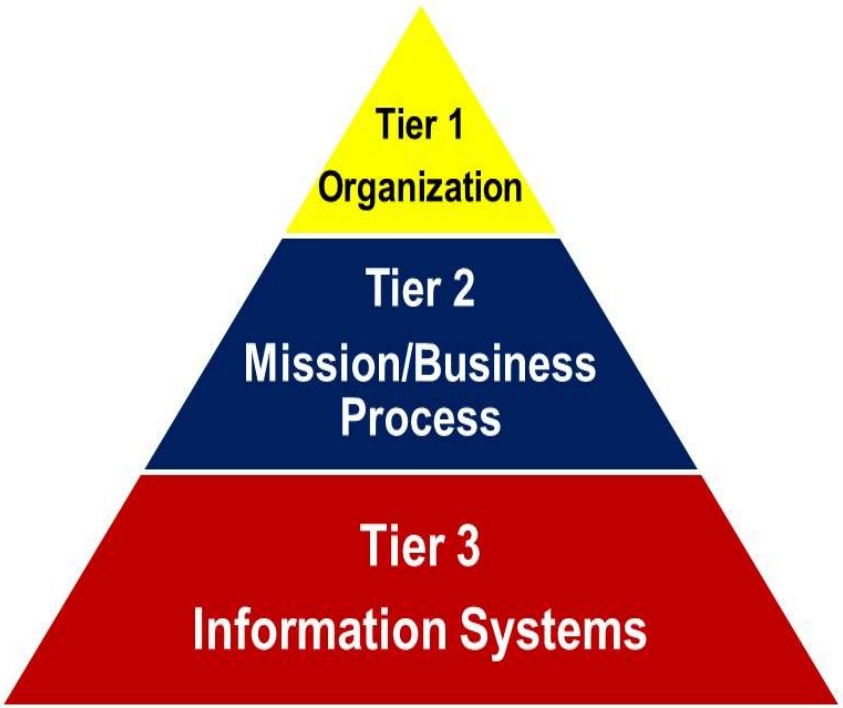
- NIST Special Publications (SP)
  - 800-37—Guide for Applying the RMF
  - 800-39—Managing Information Security Risks
  - 800-53—Security and Privacy Controls
  - 800-53A—Guide for Assessing the Security Controls
  - 800-60—Guide for Mapping Types of Information and Information Systems to Security Categories
  - 800-137—Information Security Continuous Monitoring
- Committee on National Security Systems (CNSS)
  - Instruction 1253—Security Categorization and Control Selection for National Security Systems
  - Instruction 4009—Information Assurance Glossary
  - Policy 11—National Policy Governing the Acquisition of IA and IA-Enabled IT Products

Applicability (8510.01)

- All DoD-owned IT or DoD-controlled IT that receive, process, store, display, or transmit DoD information. These technologies are broadly grouped as DoD IS, platform IT (PIT), IT services, and products. This includes IT supporting research, development, test and evaluation (T&E), and DoD-controlled IT operated by a contractor or other entity on behalf of the DoD

Risk Management and the RMF

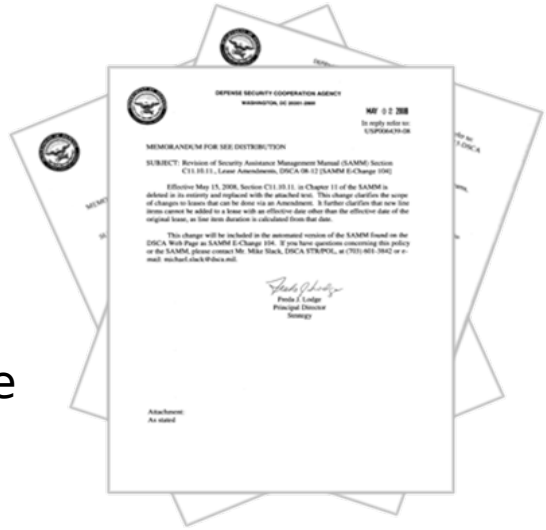
- Multi-tiered Risk Management
- DoD will implement a multi-tiered cybersecurity risk management process to protect U.S. interests, DoD operational capabilities, and DoD individuals, organizations, and assets from the DoD Information Enterprise level, through the DoD Component level, down to the IS level



Defined in NIST SP-800-39

DoD Cybersecurity Legislation, Policy and Guidance

- US Code Title 40 (Clinger-Cohen)
- Federal Information Processing Standards
- DoD Directive 8570.01, Information Assurance Training, Certification, and Workforce Management
- DoD Manual 8570.01-M, Information Assurance Workforce Improvement Program
- The National Institute of Standards and Technology
  (NIST) Special Publication (SP) 800 Series (Computer Security)
- Subchapter III of Chapter 35 of title 44, United States Code, “Federal Information Security Management Act (FISMA) of 2002”
- DoD Instruction 8500.01, Cybersecurity
- DoD Instruction 8510.01, Risk Management Framework for DoD Information Technology



Federal Information Security Management Act (FISMA)—2002

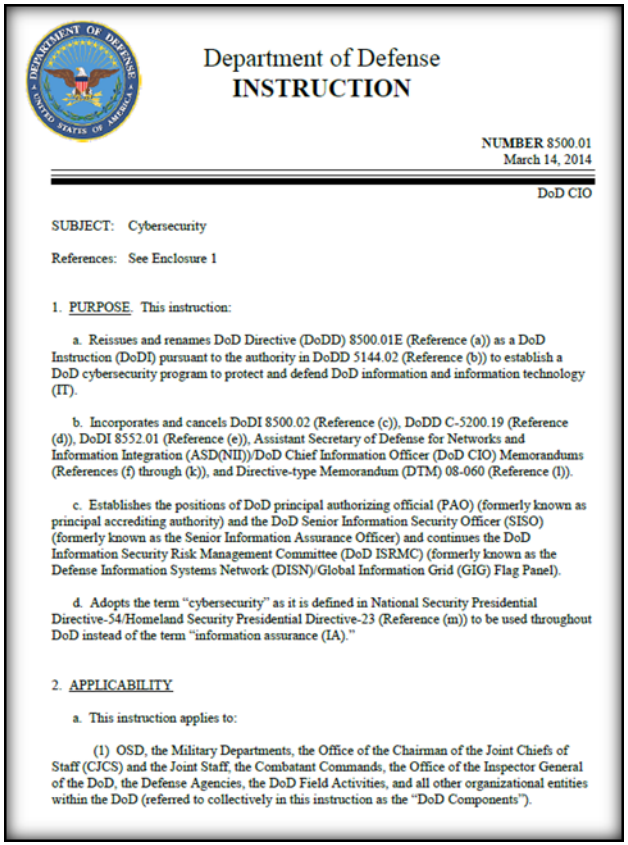
The National Institute of Standards and Technology (NIST) outlines nine steps toward compliance with FISMA:

1. Categorize the information to be protected.
2. Select minimum baseline controls.
3. Refine controls using a risk assessment procedure.
4. Document the controls in the system security plan.
5. Implement security controls in appropriate information systems.
6. Assess the effectiveness of the security controls once they have been implemented.
7. Determine agency-level risk to the mission or business case.
8. Authorize the information system for processing.
9. Monitor the security controls on a continuous basis.

The RMF must satisfy the requirements of subchapter III of chapter 35 of Title 44, United States Code (U.S.C.), also known as the "Federal Information Security Management Act (FISMA) of 2002"–*DoDI 8510.01, March 12, 2014*

DoD Cybersecurity Policy

- DoD Instruction 8500.01,Cybersecurity, Signed March 14, 2014
- Cancels or supersedes 11 DoD Directives, Instructions, or Memorandums.
- References a total of 132 policy documents
  - 12 National Institute of Standards and Technology (NIST) Special Publications
  - 9 Committee on National Security Systems (CNSS) Instructions or Policies
- Adopts the term “cybersecurity” to be used throughout the DoD instead of the term “information assurance (IA).”



Department of Defense
INSTRUCTION

NUMBER 8500.01
March 14, 2014

DoD CIO

SUBJECT: Cybersecurity

References: See Enclosure 1

1. PURPOSE. This instruction:

a. Reissues and renames DoD Directive (DoDD) 8500.01E (Reference (a)) as a DoD Instruction (DoDI) pursuant to the authority in DoDD 5144.02 (Reference (b)) to establish a DoD cybersecurity program to protect and defend DoD information and information technology (IT).

b. Incorporates and cancels DoDI 8500.02 (Reference (c)), DoDD C-5200.19 (Reference (d)), DoDI 8552.01 (Reference (e)), Assistant Secretary of Defense for Networks and Information Integration (ASD(NII))/DoD Chief Information Officer (DoD CIO) Memorandums (References (f) through (k)), and Directive-type Memorandum (DTM) 08-060 (Reference (l)).

c. Establishes the positions of DoD principal authorizing official (PAO) (formerly known as principal accrediting authority) and the DoD Senior Information Security Officer (SISO) (formerly known as the Senior Information Assurance Officer) and continues the DoD Information Security Risk Management Committee (DoD ISRMC) (formerly known as the Defense Information Systems Network (DISN)/Global Information Grid (GIG) Flag Panel).

d. Adopts the term “cybersecurity” as it is defined in National Security Presidential Directive-54/Homeland Security Presidential Directive-23 (Reference (m)) to be used throughout DoD instead of the term “information assurance (IA).”

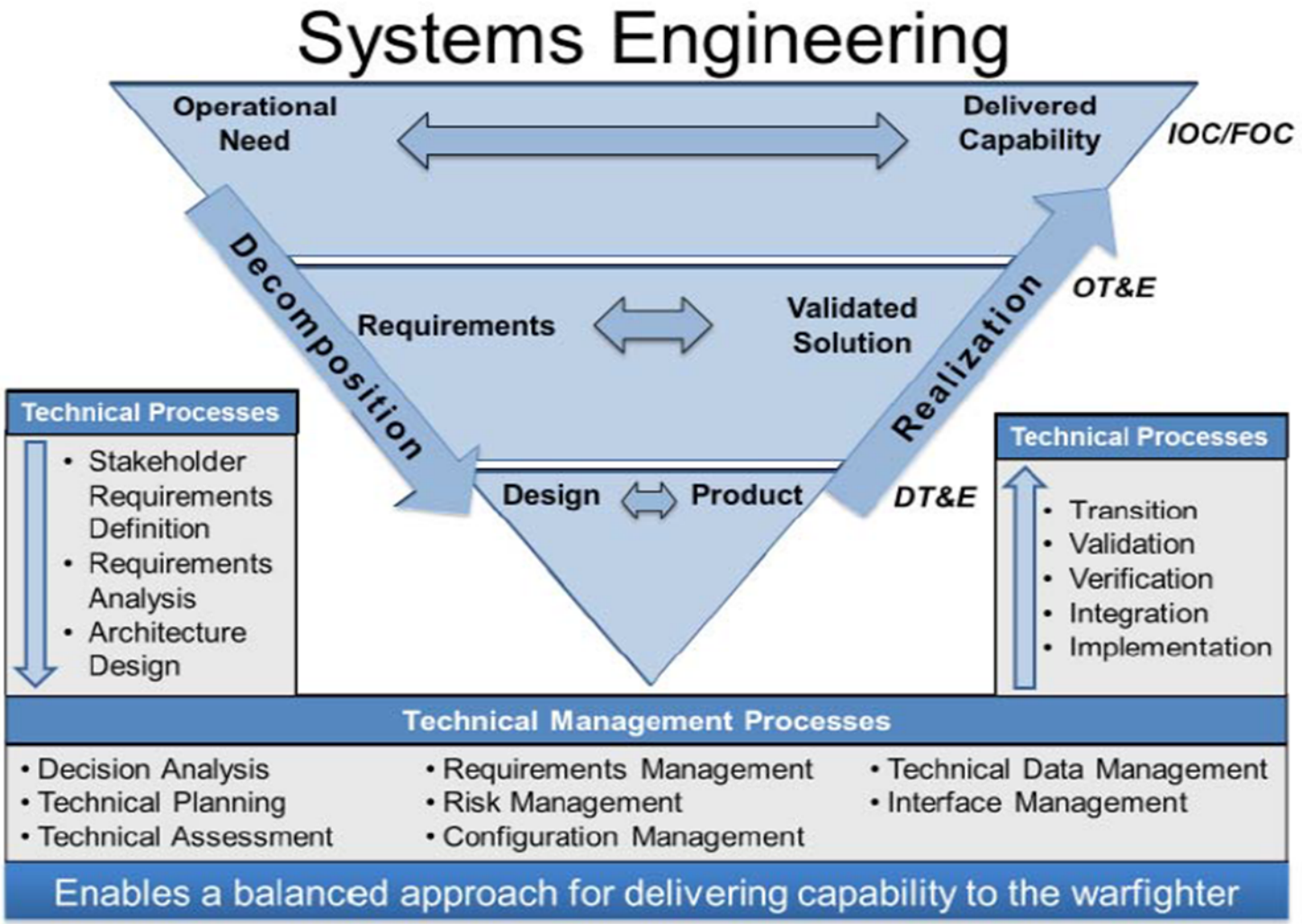
2. APPLICABILITY

a. This instruction applies to:

(1) OSD, the Military Departments, the Office of the Chairman of the Joint Chiefs of Staff (CJCS) and the Joint Staff, the Combatant Commands, the Office of the Inspector General of the DoD, the Defense Agencies, the DoD Field Activities, and all other organizational entities within the DoD (referred to collectively in this instruction as the “DoD Components”).

Link to ISA101 lesson 14.2, 14.4, and 14.5 Comp Paper#27

DoD Systems Engineering Process



Technical Management

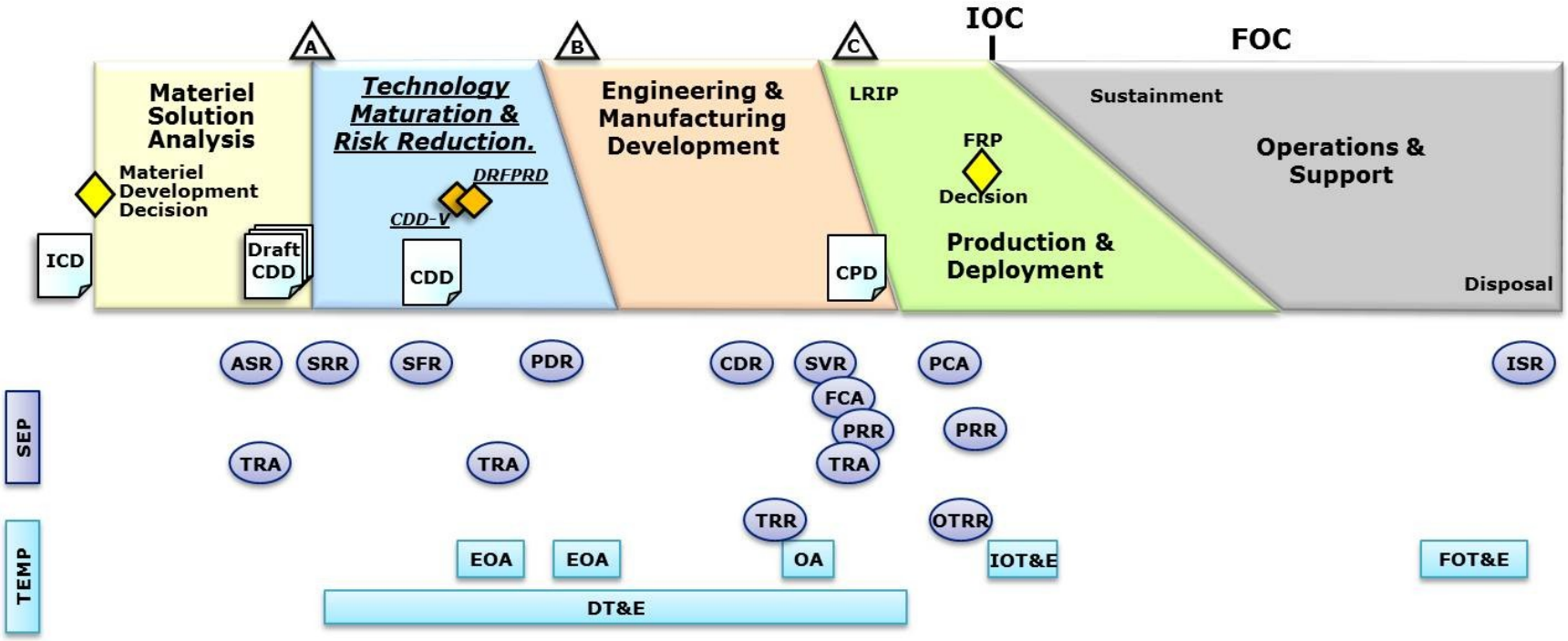
Technical Planning	Ensures that the SE processes are applied properly throughout a system's life cycle, includes defining the scope of the technical effort required to develop, field, and sustain the system, as well as providing critical quantitative inputs to program planning and life-cycle cost estimates.
Requirements Management	Ensures bi-directional traceability from the high-level requirements down to the system elements through the lowest level of the design (top down); and from any derived lower-level requirement up to the applicable source from which it originates (bottom-up).
Configuration Management	Establishes and maintains the consistency of a system's functional, performance, and physical attributes with its requirements, design, and operational documentation throughout the system's life cycle.
Interface Management	Ensure interface definition and compliance among the system elements, as well as with other systems. Documents all internal and external interface requirements and requirements changes in accordance with the program's Configuration Management Plan.
Technical Data Management	Identifies, acquires, manages, maintains, and ensures access to the technical data and computer software required to manage and support a system throughout the acquisition life cycle.
Decision Analysis	Transforms a broadly stated decision opportunity into a traceable, defendable, and actionable plan. Employ procedures, methods, and tools, such as trade studies, for identifying, representing, and formally assessing the important aspects of alternative decisions to select an optimum decision.
Technical Assessment	Compares achieved results with defined criteria to provide a fact-based understanding of the current level of product knowledge, technical maturity, program status, and technical risk. Includes methods such as technical reviews and use of technical performance measures (TPMs).
Risk Management	Involves the mitigation of program uncertainties that are critical to achieving cost, schedule, and performance goals at every stage of the life cycle. Encompasses identification, analysis, mitigation planning, mitigation plan implementation, and tracking of program risks.

Technical Processes

Stakeholder Requirements Definitions	Involves the translation of requirements from relevant stakeholders into a set of top-level technical requirements. The process helps ensure each individual stakeholder's requirements, expectations and perceived constraints are understood from the acquisition perspective.
Requirements Analysis	Involves the decomposition of top-level requirements captured by the Stakeholder Requirements Definition process into a clear, achievable, verifiable and complete set of system requirements.
Architecture Design	Involves two primary efforts; design and realization. Outputs included the detailed design down to the lowest system elements in the system architecture, and the fabrication/production procedures.
Implementation	Involves two primary efforts; detailed design and realization. Outputs included the detailed design down to the lowest system elements in the system architecture, and the fabrication/production procedures.
Integration	Incorporates the lower level system elements into a higher level system element in the physical architecture.
Verification	Provides evidence that the system or system element performs its intended functions and meets all performance requirements listed in the system performance specification and functional and allocated baselines. Verification answers the question, "Did you build the system correctly?"
Validation	Provides objective evidence that the capability provided by the system complies with stakeholder performance requirements, achieving tis use in its intended operational environment. Validation answers the question, "Is it the right solution to the problem?"
Transition	Moves any system element to the next level in the physical architecture. For the end-item system, it is the process to install and field the system to the user in the operational environment.

https://acc.dau.mil/dm

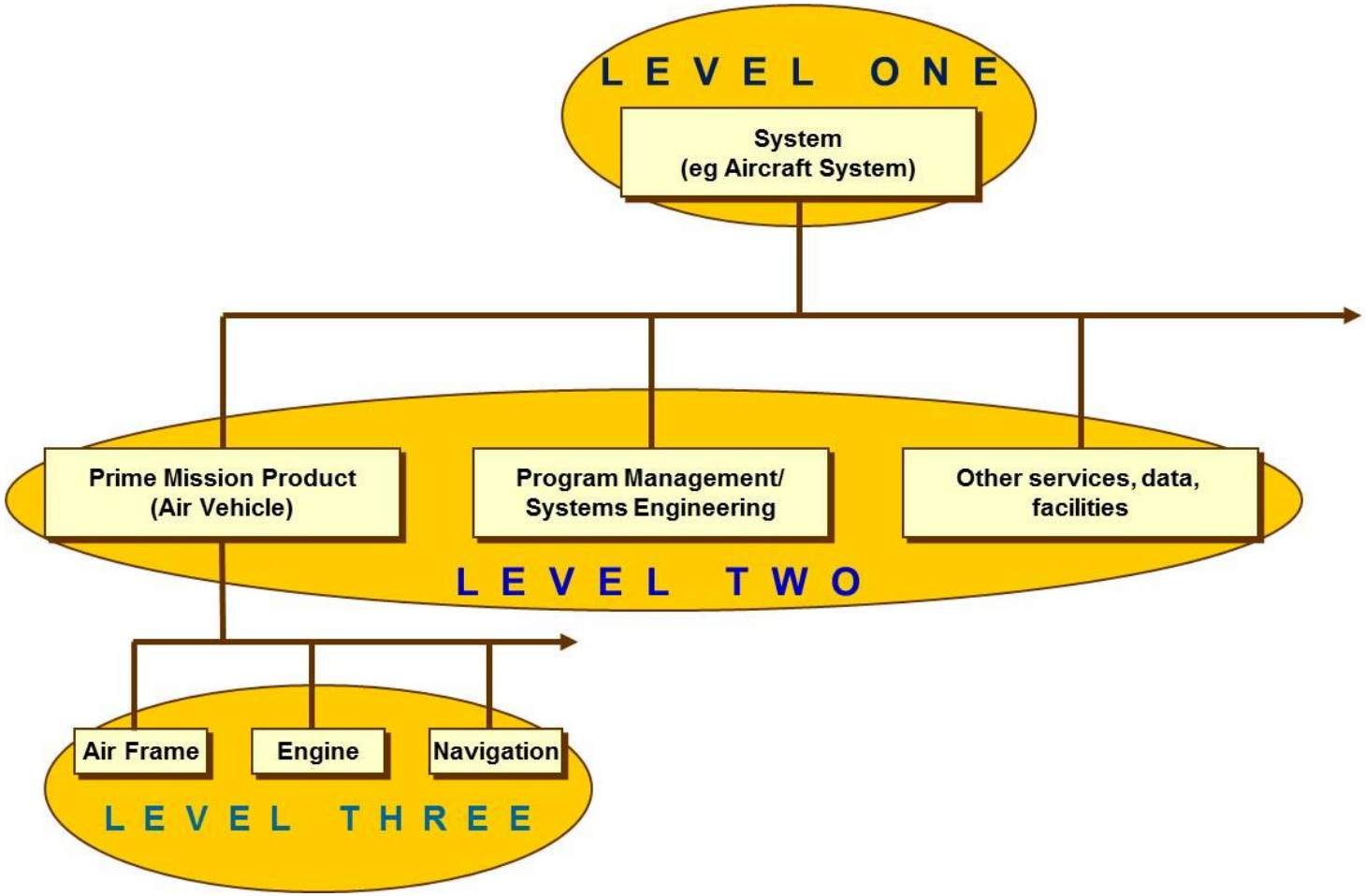
SE Technical Reviews



- Alternative Systems Review (ASR)
- Systems Requirements Review (SRR)
- System Functional Review (SFR)
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Test Readiness Review (TRR)
- System Verification Review (SVR)
- Functional Configuration Audit (FCA)
- Production Readiness Review (PRR)
- Operational Test Readiness Review (OTRR)
- Physical Configuration Audit (PCA)
- Technology Readiness Assessment (TRA)
- In-Service Review (ISR)
- Developmental Testing (DT)
- Early Operational Assessment (EOA)
- Operational Assessment (OA)
- Initial Operational Test & Evaluation (IOT&E)
- Follow on Operational Test and Evaluation (FOT&E)

Sample System Reviews

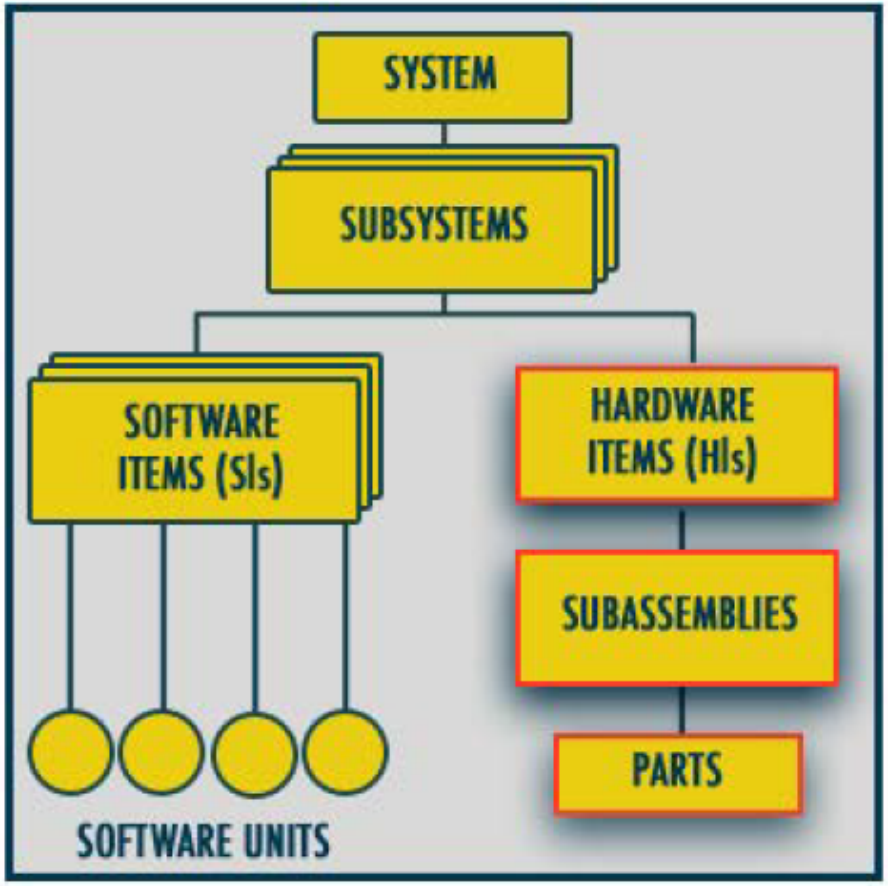
Work Breakdown Structure



Software Components in WBS

As part of the Systems Engineering Process, complex systems are decomposed into smaller subsystems and discrete products

The software requirements are broken down into Software Items (SI) and further into Software Units (SU)



Link to ISA101 lesson 19.3 and 19.4

Comp Paper#2 and #16

Modular or Flexible Contracting

- **Modular contracting**—acquisition of a system for information technology divided into several smaller acquisition increments
- Intended to:
  - Reduce program risk
  - Incentivize contractor performance
  - Meeting the Governments need for timely access to rapidly changing technology
- Each increment should utilize an appropriate contracting technique that best facilitates the acquisition of subsequent increments
- Modularity avoids the potential negative effects of one development-to-production contract by introducing check points and market competition forces a key program transition points between module contracts/contractors